ЭНЕРГЕТИКА
МИКРОЭЛЕКТРОНИКА
АВТОМАТИКА

РАСПРЕДЕЛЕННАЯ СИСТЕМА ДИСПЕТЧЕРСКОГО УПРАВЛЕНИЯ

РСДУ 5

OMS

Руководство администратора

РСДУ 5

Содержание

Система управления аварийными отключениями

Состав программного обеспечения (ПО)

В состав OMS входят следующие модули:

1. СУБД MySQL.
2. Модуль получения информации о коммутациях через очередь сообщений RabbitMQ.
3. Модуль получения информации о смене топологии сети через очередь сообщений RabbitMQ.
4. Модуль получения информации о положения автотранспорта через сервис Wialon.
5. Модуль конфигурирования OMS – omsadmin.
6. Сервер геоданных на основе Open Street Map (OSM).
7. Сервер OMS с модулем обмена данными на основе WEB API.
8. Приложения администратора для загрузки данных из внешних источников: KML файлы, системы хранения информации о потребителях и т.д.

Требования к системному ПО

1. Операционная система Ubuntu Server 64 18.04.1 и выше.
2. СУБД MySQL 8.5 и выше.
3. Python 3.7 и выше.

Требования к аппаратному обеспечению

Минимальные требования к серверу OMS: процессор Intel i7 2GHz, 4Гб RAM, 50Гб HDD. Лимитирующим фактором размера жесткого диска является запись в базу данных журнала отключенного оборудования и потребителей.

Для GIS сервера: процессор Intel i7 2GHz, 8Гб RAM, 100Гб HDD.

Порядок установки и конфигурирования

Ubuntu server

Установка ОС Ubuntu server осуществляется стандартно согласно документации производителя ОС.

Все действия со стороны OMS производятся под пользователем omsserver. Необходимо создать этого пользователя со следующими характеристиками:

```
omsserver:x:[UID]:[GID]::/var/www/:/bin/false
```

GID – www-data

СУБД MySQL

Установка MySQL осуществляется стандартно из репозитария Ubuntu Server. Пароль пользователя [PASSWORD] необходимо заменить на новый.

```
sudo apt update
sudo apt install mysql-server
sudo mysql_secure_installation
```

Создание схемы данных pathfinder для OMS:

```
sudo mysql -u root -p
```

Для MySQL 5.7:

```
create database pathfinder;
GRANT ALL PRIVILEGES ON *.* TO 'pathfinder'@'%'
IDENTIFIED BY '[PASSWORD];
```

Для MySQL 8:

```
create database pathfinder;
CREATE USER 'pathfinder'@'%' IDENTIFIED BY
'[PASSWORD]';
GRANT ALL ON pathfinder.* TO 'pathfinder'@'%';
```

После этого необходимо загрузить схему данных, например, с помощью приложения MySQLWorkbench или консольного приложения:

```
sudo mysql -u root -p < OmsDbInstall.sql
```

Nginx

Web сервер Nginx устанавливается стандартно:

```
sudo apt install nginx
```

После этого необходимо создать файл конфигурации `/etc/nginx/sites-available/omsserver.conf` со следующими параметрами:

```
upstream omsserver {
    server unix:/tmp/omsserver-uwsgi.sock;
}

server {
    listen 80;
    server_tokens off;
    server_name [IP адрес сервера];

    location / {
        root /var/www/omsfrontend/dist;

      location ~* \.html$ {
            add_header Cache-Control "no-store, no-
cache, must-revalidate";
        }

        location ~*
\.(?:js|css|ico|png|svg|gif|jpe?g|woff|woff2|eot|ttf|
otf)$ {
            etag off;
            add_header Cache-Control "private, max-
age=31536000, immutable";
            access_log off;
        }
    }
    location /socket.io {
        include proxy_params;
        proxy_http_version 1.1;
        proxy_buffering off;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_pass http://127.0.0.1:5000/socket.io;
    }
```

```
    location /rsdu/oms/api {
        include proxy_params;
        proxy_pass http://127.0.0.1:5000;
        uwsgi_pass_header Session-Key;
    }

    location /static {
        root /var/www/omsserver;
    }

    location /rsdu/oms/static {
        root /var/www/omsserver;
    }
}
```

Строку [IP адрес сервера] необходимо заменить на реальный адрес сервера.

Выполнить команду:

```
ln    -s    /etc/nginx/sites-available/omsserver.conf
/etc/nginx/sites-enabled/
```

Создать каталог:

```
mkdir /var/www/omsserver
chown -R omsserver:www-data /var/www/omsserver
```

Далее повторяем действия для приложения администрирования. Создаем файл `/etc/nginx/sites-available/omasadmin.conf`:

```
upstream uwsgi_omsadmin_upstream {
    server unix:/tmp/omsadmin-uwsgi.sock;
}

server {
    listen 8080;
    server_tokens off;
    server_name [IP адрес сервера];

    location / {
        include proxy_params;
        proxy_pass http://127.0.0.1:5005;
        uwsgi_pass_header Session-Key;
    }

    location /static {
        root /var/www/omsadmin;
    }

    location /rsdu/oms/static {
        root /var/www/omsadmin;
    }
}
```

Строку [IP адрес сервера] необходимо заменить на реальный адрес сервера.

Выполнить команду:

```
ln -s /etc/nginx/sites-available/omsadmin.conf
/etc/nginx/sites-enabled/
```

Создать каталог:

```
mkdir /var/www/omsadmin
chown -R omsserver:www-data /var/www/omsadmin
```

Python

Установка языка Python производится стандартным образом из репозитария Ubuntu Server:

```
sudo apt-get install python3.6
sudo apt-get -y install python3-pip
sudo pip3 install --upgrade pip
sudo apt install -y python3-venv
```

Далее создается виртуальное окружение в каталогах `/var/www/omsserver` и `/var/www/omsadmin`:

```
python3 -m venv /var/www/omsadmin/venv
python3 -m venv /var/www/omsserver/venv
```

В каталоги `/var/www/omsserver` и `/var/www/omasdamin` помещается содержимое архивов с ПО сервера OMS и устанавливаются необходимые пакеты Python:

```
cd /var/www/omssserver
source venv/bin/activate

pip install -r requirements.txt
```

```
cd /var/www/omssserver
source venv/bin/activate

pip install -r requirements.txt
```

После этого создаются файлы запуска сервисов systemd:

/etc/systemd/system/omsserver.service

```
[Unit]
Description = RsduOmsServer
After = mysql.service

[Service]
PermissionsStartOnly = true
PIDFile = /run/omsserver/omsserver.pid
User = omsserver
Group = www-data
WorkingDirectory = /var/www/omsserver

ExecStartPre = /bin/mkdir /run/omsserver
ExecStartPre = /bin/chown -R omsserver:www-data /run/omsserver
ExecStartPre = /bin/chown omsserver:www-data /var/log/omsserver.log
ExecStartPre = /bin/chown -R omsserver:www-data /var/www/omsserver
ExecStart = /var/www/omsserver/venv/bin/gunicorn wsgi:app --worker-class gevent -w 1 --bind 127.0.0.1:5000 --pid /run/omsserver/omsserver.pid --access-logfile /var/log/omsserver.log
ExecReload = /bin/kill -s HUP $MAINPID
ExecStop = /bin/kill -s TERM $MAINPID
ExecStopPost = /bin/rm -rf /run/omsserver
ExecStopPost = /bin/rm -rf /var/www/omsserver/__pycache__
PrivateTmp = true

[Install]
WantedBy = multi-user.target
```

/etc/systemd/system/omsadmin.service:

```
[Unit]
Description = RsduOmsAdmin
After = mysql.service

[Service]
PermissionsStartOnly = true
PIDFile = /run/omsadmin/omsadmin.pid
User = omsserver
Group = www-data
WorkingDirectory = /var/www/omsadmin

ExecStartPre = /bin/mkdir /run/omsadmin
ExecStartPre = /bin/chown -R omsserver:www-data /run/omsadmin
ExecStartPre = /bin/chown omsserver:www-data /var/log/omsadmin.log
ExecStartPre = /bin/chown -R omsserver:www-data /var/www/omsadmin
ExecStart = /var/www/omsadmin/venv/bin/gunicorn wsgi:app --worker-class
gevent -w 1 --bind 127.0.0.1:5005 --pid /run/omsadmin/omsadmin.pid --access-
logfile /var/log/omsadmin.log
ExecReload = /bin/kill -s HUP $MAINPID
ExecStop = /bin/kill -s TERM $MAINPID
ExecStopPost = /bin/rm -rf /run/omsadmin
ExecStopPost = /bin/rm -rf /var/www/omsadmin/__pycache__
PrivateTmp = true

[Install]
WantedBy = multi-user.target
```

Настраивается запуск сервисов:

```
systemctl enable omsadmin.service
touch /var/log/omsadmin.log
systemctl start omsadmin.service
```

```
systemctl enable omsserver.service
touch /var/log/ omsserver.log
systemctl start omsserver.service
```

Модуль получения информации о коммутациях через очередь сообщений RabbitMQ

Назначение этого модуля – получение информации о коммутациях и передача ее через WEB API непосредственно в сервер OMS.

Месторасположение модуля: `/usr/lib/ema/oms/omsrgw`

Исполняемый файл модуля: `oms-gw-rmq-signal.py`

Конфигурационный файл модуля: `oms-gw-rmq-signal.cfg`

Лог файл: `/var/log/oms-gw-rmq-signal.log`

```
[RMQ]
; HOST or HOST1,HOST2,..
HOST = 10.5.165.20,10.5.165.30
PORT = 5672
VIRTUAL_HOST = rsdu
CONNECTION_NAME = RSDU OMS Gateway Signal
EXCHANGE = rsdu.signal
LOGIN = rsdu
PASSWORD = [PASSWORD]

[OMS]

NOTIFY_URL =
http://127.0.0.1/rsdu/oms/api/omsgw/signal

[LOG]
; INFO, DEBUG
LEVEL = DEBUG
```

Запуск модуля из `/etc/systemd/system/oms-gw-rmq-signal.service`:

```
[Unit]
Description=RSDU OMS RabbitMQ signal gateway
After=mysql.service

[Service]
Type=simple
User=root
Group=root
WorkingDirectory=/usr/lib/ema/oms/omsrgw
ExecStart=/usr/lib/ema/oms/omsrgw/oms-gw-rmq-signal.py
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
```

Установка виртуального окружения аналогична установки для omsserver.

Модуль получения информации о смене топологии сети через очередь сообщений RabbitMQ

Назначение этого модуля – получение информации о смене топологии и передача ее через WEB API непосредственно в сервер OMS.

Месторасположение модуля: `/usr/lib/ema/oms/omsrgw`

Исполняемый файл модуля: `oms-gw-rmq-topology.py`

Конфигурационный файл модуля: `oms-gw-rmq-topology.cfg`

Лог файл: `/var/log/oms-gw-rmq-topology.log`

```
[RMQ]
; HOST or HOST1,HOST2,..
HOST = 10.5.165.20,10.5.165.30
PORT = 5672
VIRTUAL_HOST = rsdu
CONNECTION_NAME = RSDU OMS Gateway Topology

EXCHANGE = rsdu.events
ROUTING_KEY =
rsdu.scada.topology.NETWORK_STATE_UPDATE

LOGIN = rsdu
PASSWORD = [PASSWORD]

[OMS]

NOTIFY_URL =
http://127.0.0.1/rsdu/oms/api/omsgw/topochanged

[LOG]
; INFO, DEBUG
LEVEL = DEBUG
```

Запуск модуля из `/etc/systemd/system/oms-gw-rmq-topology.service`:

```
[Unit]
Description=RSDU OMS RabbitMQ topology gateway
After=mysql.service

[Service]
Type=simple
User=root
Group=root
WorkingDirectory=/usr/lib/ema/oms/omsrgw
ExecStart=/usr/lib/ema/oms/omsrgw/oms-gw-rmq-topology.py
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
LEVEL = INFO
```

Модуль получения информации о положения автотранспорта через сервис Wialon

Назначение этого модуля – получение информации о положении автотранспорта с портала Wialon и передача ее через WEB API непосредственно в сервер OMS.

Месторасположение модуля: `/usr/lib/ema/oms/omsrgw`

Исполняемый файл модуля: `oms-gw-wialon-locations.py`

Конфигурационный файл модуля: `oms-gw-wialon-locations.cfg`

Лог файл: `/var/log/oms-gw-wialon-locations.log`

```
[WIALON]

HOST = [АДРЕС СЕРВЕРА WIALON]
TOKEN = [ТОКЕН ДЛЯ ДОСТУПА]
PERIOD_SEC = 180
SAVED_DEPTH_PERIODS = 120

[MYSQL]

MYSQL_DATABASE_HOST = 127.0.0.1
MYSQL_DATABASE_USER = pathfinder
MYSQL_DATABASE_PASSWORD = [ПАРОЛЬ ПОЛЬЗОВАТЕЛЯ БД]
MYSQL_DATABASE_DB = pathfinder

[LOG]
; INFO, DEBUG
LEVEL = INFO
```

Запуск модуля из `/etc/systemd/system/oms-gw-wialon-locations.service`:

```
[Unit]
Description=RSDU OMS Wialon locations gateway
After=mysql.service

[Service]
Type=simple
User=root
Group=root
WorkingDirectory=/usr/lib/ema/oms/omsrgw
ExecStart=/usr/lib/ema/oms/omsrgw/oms-gw-wialon-locations.py
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
```

Модуль конфигурирования OMS

Модуль позволяет осуществлять конфигурирование системы OMS с помощью WEB интерфейса. Доступ к модулю осуществляется по адресу `http://[IP адрес сервера OMS]:8080`

Модуль позволяет настраивать следующие разделы базы данных сервера OMS:

1. Потребители
2. Бригады
3. Сотрудники
4. Отключения
5. Оборудование
6. Справочники
7. Геоинформационная система (GIS)